Problem Statement:

Right now the System.Catalog table can't split due to the locking while propagating view metadata. This locking is required because while modifying a view the parent table can mutate thus putting everything in an inconsistent state.

Proposal:

Only store the metadata about the view itself and deal with the propagation at read time. Thus if you have a Parent Table with columns A, B, C and a view where you add column D. Only store that respective information on disk and resolve that the View has columns A,B,C,D at read time.

MetadataEndpointImpl is server-side logic and if we can do all the work for this feature on the server we can complete it in one release.

Things we have to take care of:

1. Do not propagate view metadata to children.
2. Ensure that the server-side cache does not change - we only change what we return to the client. Thus the PTable for a view is only its own inherent metadata and nothing from its parent.
3. When returning this result to the client, make sure we combine the parent metadata for views such that the client sees no difference in what it was receiving from an older release.
4. Push the combining of indexing logic back onto the server, no reason to have this in the client and make redundant scan requests.
5. Upgrade should happen before the first server goes live, we will have to change how view columns are stored, remove all columns that are inherited from the parent.
6. Drop columns do not have to propagate
7. Drop views will have to delete the view and maybe we delete if / whenever we try and read children. Still have to figure this one out.